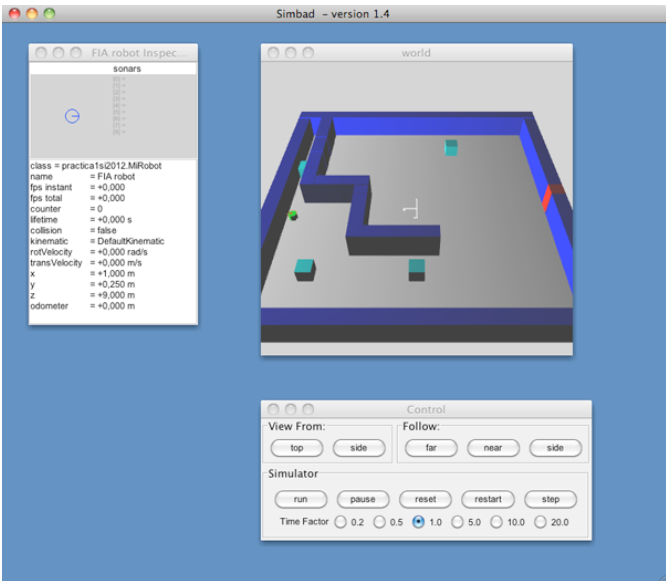
# Lógica Difusa

INTRODUCCIÓN

El sistema experto difuso que se debe realizar debe seguir el algoritmo A\* calculado en la práctica anterior y evitar obstáculos utilizando los sensores y actuadores del robot.

En la práctica se utilizará el lenguaje FCL (Fuzzy Controller Language). Este lenguaje es prácticamente una transcripción del lenguaje natural para la especificación de sistemas difusos.



REALIZACIÓN DE LA PRÁCTICA

Empezamos la práctica con el fichero *controller.fcl* por defecto. En este caso el Robot no hace nada coherente con el A\*. A continuación se irá detallando los distintos estados por donde ha pasado la práctica:

# Prueba 1

## Variables

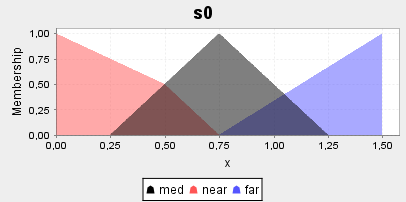
Defino los sensores s0, s1 y s8 como:

FUZZIFY s0

TERM near := (0,1) (0.5,0.5) (0.75,0);

TERM med := (0.25,0) (0.75,1) (1.25,0) (1.5,0);

TERM far := (0.75,0) (1.5,1);

END\_FUZZIFY

Y para la variable sig como:

FUZZIFY sig

TERM izqm := (-180,0) (0,0) (180,1);

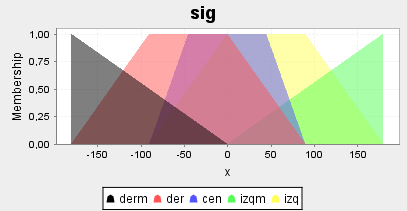
TERM izq := (-180,0)(-90,0) (0,1) (90,1) (180,0);

TERM cen := (-180,0) (-90,0) (-45,1) (45,1) (90,0);

TERM der := (-180,0) (-90,1) (0,1) (90,0) (180,0);

TERM derm := (-180,1) (0,0) (180,0);

END\_FUZZIFY



## Reglas

RULE 1: IF s0 IS far THEN vel IS fast;

RULE 10: IF s0 IS med THEN vel IS med;

RULE 2: IF s0 IS near AND s1 IS far THEN rot IS derm;

RULE 12: IF s0 IS near AND s1 IS med THEN rot IS der;

RULE 13: IF s0 IS near AND s8 IS med THEN rot IS izq;

RULE 7: IF s0 IS near AND s8 IS far THEN rot IS izqm;

RULE 8: IF sig IS izq THEN rot IS izqm;

RULE 9: IF sig IS der THEN rot IS derm;

RULE 11: IF sig IS cen THEN rot IS cen;

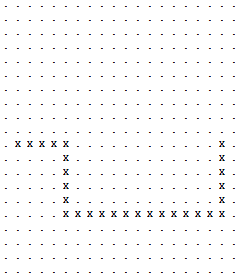
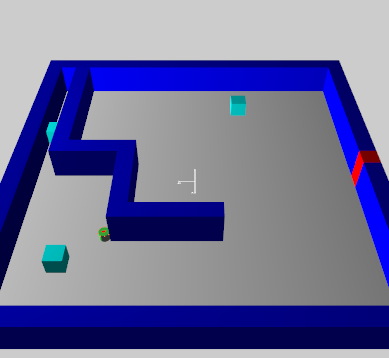
RULE 3: IF s1 IS near THEN rot IS der;

RULE 4: IF s8 IS near THEN rot IS izq;

RULE 5: IF s1 IS far THEN vel IS med;

RULE 6: IF s8 IS far THEN vel IS med;

## Resultado



Sigue el A\* hasta la esquina con la pared, donde el robot colisiona por su derecha. El sensor más cercano a la pared es s1 y s2.

# Prueba 2

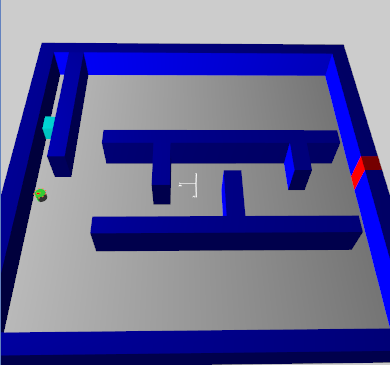
## Reglas

He añadido las siguientes reglas:

RULE 4: IF s8 IS near THEN rot IS izq;

RULE 3: IF s2 IS near THEN rot IS der;

En la prueba anterior el robot colisionaba en los extremos porque se acercaba mucho a la pared. Añadiendo las reglas anteriores se consigue que no colisione.

Cambiado a mapas más complejos como el siguiente no era capaz de recorrerlo.

Así que me he replanteado todo. Desde los conjuntos.

# Prueba 3

## Variables

Defino s0, s1, s2, s7, y s8 como:

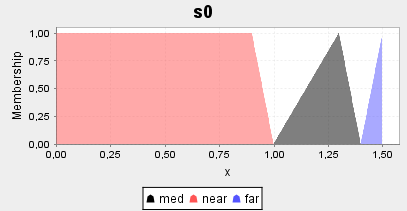
FUZZIFY s0

TERM near := (0,1) (0.5,1) (0.9,1) (1,0);

TERM med := (1,0)(1.3,1)(1.4,0);

TERM far := (1.4,0) (1.5,1);

END\_FUZZIFY



Además de cambiar los conjuntos de los sensores he cambiado también el de siguiente:

FUZZIFY sig

TERM izqm:=(-180,0)(90,0)(135,1)(180,1);

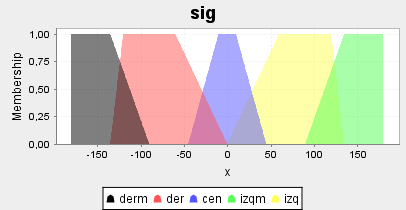
TERM izq :=(-180,0)(0,0)(60,1)(120,1)(135,0)(180,0);

TERM cen:=(-180,0)(-45,0)(-10,1)(10,1)(45,0)(180,0);

TERM der:=(-180,0)(-135,0)(-120,1)(-60,1)(0,0) (180,0);

TERM derm:=(-180,1)(-135,1)(-90,0)(180,0);

END\_FUZZIFY



Y también he cambiado las variables de vel y rot:

La variable rot la he definido desde –π hasta π:

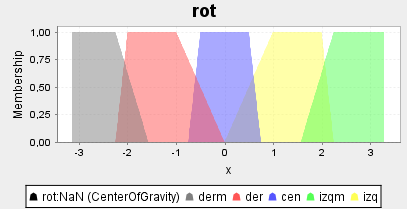
TERM izqm :=(-3.14,0) (1.57,0) (2.25,1) (3.14,1);

TERM izq :=(-3.14,0)(0,0)(1,1)(2.0,1)(2.25,0)(3.14,0);

TERM cen:=(-3.14,0)(-0.75,0)(-0.5,1)(0.5,1)(0.75,0)(3.14,0);

TERM der:=(-3.14,0)(-2.25,0)(-2.0,1)(-1,1)(0,0)(3.14,0);

TERM derm:=(-3.14,1)(-2.25,1)(-1.57,0)(3.14,0);

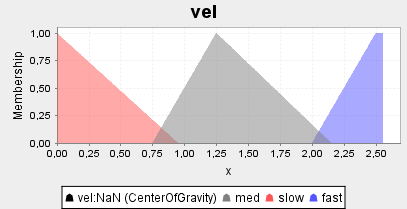


Y la variable de la velocidad como:

TERM fast := (2,0) (2.5,1);

TERM med := (0.75,0) (1.25,1) (2.15,0);

TERM slow := (0,1) (0.95,0);



## Reglas

Estas reglas controlan que el Robot siga el camino, girando hacia sig.

RULE 50: IF sig IS izqm AND s1 IS far THEN rot IS izqm;

RULE 51: IF sig IS izqm AND s1 IS med THEN rot IS izqm;

RULE 52: IF sig IS derm AND s8 IS far THEN rot IS derm;

RULE 53: IF sig IS derm AND s8 IS med THEN rot IS derm;

RULE 57: IF sig IS izq AND s1 IS far THEN rot IS izq;

RULE 58: IF sig IS izq AND s1 IS med THEN rot IS izq;

RULE 59: IF sig IS der AND s8 IS far THEN rot IS der;

RULE 60: IF sig IS der AND s8 IS med THEN rot IS der;

RULE 54: IF sig IS cen AND s0 IS far THEN rot IS cen;

RULE 55: IF sig IS cen AND s0 IS med THEN rot IS cen;

Estas reglas controlan que el Robot no colisiones contra las paredes por los laterales.

RULE 40: IF s1 IS near THEN rot IS derm;

RULE 41: IF s8 IS near THEN rot IS izqm;

RULE 42: IF s1 IS med THEN rot IS der;

RULE 43: IF s8 IS med THEN rot IS izq;

RULE 45: IF s2 IS near THEN rot IS derm;

RULE 46: IF s7 IS near THEN rot IS izqm;

RULE 47: IF s2 IS med THEN rot IS der;

RULE 48: IF s7 IS med THEN rot IS izq;

RULE 44: IF s1 IS far AND s8 IS far THEN rot IS cen;

Y por último con estas reglas controlo la velocidad del Robot.

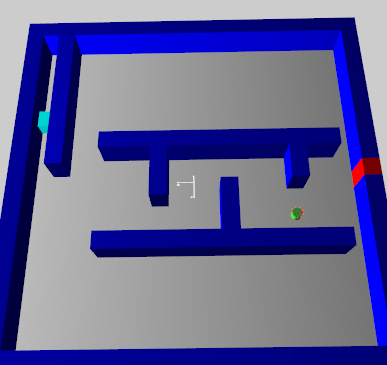
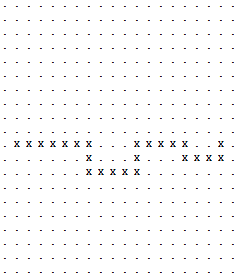
RULE 1: IF s0 IS far THEN vel IS med;

RULE 2: IF s0 IS med THEN vel IS slow;

RULE 3: IF s0 IS near THEN vel IS slow;

## Resultados

Ante un mundo más complicado como el anterior el Robot consigue llegar a su destino.



Con estas reglas el Robot ha conseguido pasar todos los mapas que le he pasado. Son mapas que tienen una solución viable.